Red Hat Advanced Cluster Management for Kubernetes 2.10

ネットワーク

ネットワーク

Last Updated: 2024-05-24

Red Hat Advanced Cluster Management for Kubernetes 2.10 ネットワーク

ネットワーク

法律上の通知

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at
http://creativecommons.org/licenses/by-sa/3.0/
. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

ハブクラスターとマネージドクラスターのネットワーク要件について詳しく説明しています。

目次

第1章 ネットワーク **3**
1.1. ハブクラスターのネットワーク設定 3
1.2. マネージドクラスターのネットワーク設定 5
1.3. 高度なネットワーク設定 7
1.4. SUBMARINER マルチクラスターネットワーキングおよびサービスディスカバリー 10

第1章 ネットワーク

ここでは、ハブクラスターとマネージドクラスターの両方のネットワーク要件を説明します。

- ハブクラスターのネットワーク設定
- マネージドクラスターのネットワーク設定
- 高度なネットワーク設定
- Submariner マルチクラスターネットワーキングおよびサービスディスカバリー

1.1. ハブクラスターのネットワーク設定

重要: 信頼された CA バンドルは Red Hat Advanced Cluster Management の namespace で利用できますが、その拡張にはネットワークへの変更が必要です。信頼できる CA バンドル ConfigMap は、**trusted-ca-bundle** のデフォルト名を使用します。この名前は、**TRUSTED_CA_BUNDLE** という名前の環境変数で Operator に提供すると変更できます。詳細は、Red Hat OpenShift Container Platform の **ネットワーク** セクションの クラスター全体のプロキシーの設定 を参照してください。

ハブクラスターネットワークの設定を参照できます。

1.1.1. ハブクラスターのネットワーク設定表

次の表でハブクラスターネットワーク要件を参照してください。

方向	プロトコル	接続	ポート (指定されている場合)	送信元アドレス	宛先アドレス
マネージドクラスターへのアウトバウンド	HTTPS	マネージドクラスターの Pod のログを Search コンソールから動的に取得し、マネージドクラスターで実行している **klusterlet-addon-workmgr** サービスを使用します。	443	なし	マネージドクラスタールートにアクセスするための IP アドレス

方向	プロトコル	接続	ポート (指定されている場合)	送信元アドレス	宛先アドレス
マネージドクラスターへのアウトバウンド	HTTPS	klusterlet をインストールするためにインストール時にプロビジョニングされるマネージドクラスターの Kubernetes API サーバー	6443	なし	Kubernetes マネージドクラスター API サーバーの IP
チャネルソースへの送信	HTTPS	アプリケーションライフサイクル、OpenShift GitOps、または Argo CD を使用して接続する場合にのみ必要となる、GitHub、Object Store、および Helm リポジトリーを含むチャネルソース	443	なし	チャネルソースの IP
マネージドクラスターからの受信	HTTPS	メトリクスおよびアラートをプッシュするマネージドクラスターは、OpenShift Container Platform バージョン 4.13 以降を実行するマネージドクラスターに対してのみアラートが収集されます	443	なし	ハブクラスターアクセスルートへの IP アドレス
マネージドクラスターからの受信	HTTPS	マネージドクラスターからの変更を監視するハブクラスターの Kubernetes API サーバー	6443	なし	ハブクラスター Kubernetes API サーバーの IP アドレス

方向	プロトコル	接続	ポート (指定されている場合)	送信元アドレス	宛先アドレス
ObjectStore へのアウトバウンド	HTTPS	Cluster Backup Operator の実行時に、長期保存用の可観測性メトリクスデータを送信します	443	なし	ObjectStore の IP アドレス
イメージリポジトリーへのアウトバウンド	HTTPS	OpenShift Container Platform および Red Hat Advanced Cluster Management のイメージにアクセスします。	443	なし	イメージリポジトリーの IP アドレス

1.2. マネージドクラスターのネットワーク設定

マネージドクラスターネットワークの設定を参照できます。

1.2.1. マネージドクラスターのネットワーク設定表

次の表でマネージドクラスターネットワーク要件を参照してください。

方向	プロトコル	接続	ポート (指定されている場合)	送信元アドレス	宛先アドレス
ハブクラスターからの受信	HTTPS	マネージドクラスターの Pod の Search コンソールからログを動的に送信するには、マネージドクラスターで実行している **klusterlet-addon-workmgr** サービスを使用します。	443	なし	マネージドクラスタールートにアクセスするための IP アドレス

方向	プロトコル	接続	ポート (指定されている場合)	送信元アドレス	宛先アドレス
ハブクラスターからの受信	HTTPS	klusterlet をインストールするためにインストール時にプロビジョニングされるマネージドクラスターの Kubernetes API サーバー	6443	なし	Kubernetes マネージドクラスター API サーバーの IP
イメージリポジトリーへのアウトバウンド	HTTPS	OpenShift Container Platform および Red Hat Advanced Cluster Management のイメージにアクセスします。	443	なし	イメージリポジトリーの IP アドレス
ハブクラスターへの送信	HTTPS	メトリクスおよびアラートをプッシュするマネージドクラスターは、OpenShift Container Platform バージョン 4.13 以降を実行するマネージドクラスターに対してのみアラートが収集されます	443	なし	ハブクラスターアクセスルートへの IP アドレス
ハブクラスターへの送信	HTTPS	ハブクラスターの Kubernetes API サーバーで変更の有無を監視します。	6443	なし	ハブクラスター Kubernetes API サーバーの IP アドレス

方向	プロトコル	接続	ポート (指定されている場合)	送信元アドレス	宛先アドレス
チャネルソースへの送信	HTTPS	アプリケーションライフサイクル、OpenShift GitOps、または Argo CD を使用して接続する場合にのみ必要となる、GitHub、Object Store、および Helm リポジトリーを含むチャネルソース	443	なし	チャネルソースの IP

1.3. 高度なネットワーク設定

- Infrastructure Operator の追加のネットワーク要件表
- Submariner のネットワーク要件表
- Hive テーブルの追加のネットワーク要件表
- ホステッドコントロールプレーンのネットワーク要件表 (テクノロジープレビュー)
- アプリケーションデプロイメントのネットワーク要件表
- namespace 接続のネットワーク要件表

1.3.1. Infrastructure Operator の追加のネットワーク要件表

Infrastructure Operator を使用してベアメタルマネージドクラスターをインストールする場合、追加のネットワーク要件については、Kubernetes Operator ドキュメントのマルチクラスターエンジンの ネットワーク設定 を参照してください。

1.3.2. Submariner のネットワーク要件表

Submariner を使用するクラスターに対して、ポートを 3 つ開放する必要があります。以下の表は、どのポートを使用できるかを示しています。

方向	プロトコル	接続	ポート (指定されている場合)
送信および受信	UDP	各マネージドクラスター	4800

方向	プロトコル	接続	ポート (指定されている場合)
送信および受信	UDP	各マネージドクラスター	4500、500、およびゲートウェイノード上の IPsec トラフィックに使用されるその他のポート
受信	TCP	各マネージドクラスター	8080

1.3.3. Hive テーブルの追加のネットワーク要件表

Central Infrastructure Management の使用が含まれる Hive Operator を使用してベアメタルマネージドクラスターをインストールする場合は、ハブクラスターと **libvirt** プロビジョニングホスト間で、レイヤー 2 またはレイヤー 3 のポート接続を設定する必要があります。プロビジョニングホストへのこの接続は、Hive を使用したベースベアメタルクラスターの作成時に必要になります。詳細は、以下の表を参照してください。

方向	プロトコル	接続	ポート (指定されている場合)
libvirt プロビジョニングホストへのハブクラスターの送信および受信	IP	Hive Operator がインストールされているハブクラスターを、ベアメタルクラスターの作成時にブートストラップとして機能する **libvirt** プロビジョニングホストに接続します。	

注記: これらの要件はインストール時にのみ適用され、Infrastructure Operator でインストールされたクラスターのアップグレード時には必要ありません。

1.3.4. ホステッドコントロールプレーンのネットワーク要件表 (テクノロジープレビュー)

ホステッドコントロールプレーンを使用する場合、**HypershiftDeployment** リソースには、次の表に示すエンドポイントへの接続が必要です。

方向	接続	ポート (指定されている場合)
Outbound	OpenShift Container Platform コントロールプレーンおよびワーカーノード	
Outbound	Amazon Web Services のホステッドクラスターのみ: AWS API および S3 API へのアウトバウンド接続	

方向	接続	ポート (指定されている場合)
Outbound	Microsoft Azure クラウドサービスのホステッドクラスターのみ: Azure API へのアウトバウンド接続	
Outbound	coreOS の ISO イメージと OpenShift Container Platform Pod のイメージレジストリーを格納する OpenShift Container Platform イメージリポジトリー	
Outbound	ホスティングクラスター上の klusterlet のローカル API クライアントは、HyperShift がホストするクラスターの API と通信します。	

1.3.5. アプリケーションデプロイメントのネットワーク要件表

一般的なアプリケーションのデプロイメント通信は、マネージドクラスターからハブクラスターへの一方向です。接続では、マネージドクラスターのエージェントによって設定される **kubeconfig** を使用します。マネージドクラスターでのアプリケーションデプロイメントは、ハブクラスターの以下の namespace にアクセスする必要があります。

- チャネルリソースの namespace
- マネージドクラスターの namespace

1.3.6. namespace 接続のネットワーク要件表

- アプリケーションライフサイクル接続:
  - namespace の **open-cluster-management** は、ポート 4000 のコンソール API にアクセスする必要があります。
  - namespace の **open-cluster-management** は、ポート 3001 でアプリケーション UI を公開する必要があります。
- アプリケーションライフサイクルバックエンドコンポーネント (Pod):
  ハブクラスターでは、以下の Pod を含む **open-cluster-management** namespace にすべてのアプリケーションライフサイクル Pod がインストールされます。
  - multicluster-operators-hub-subscription
  - multicluster-operators-standalone-subscription
  - multicluster-operators-channel
  - multicluster-operators-application
  - multicluster-integrations

これらの Pod が **open-cluster-management** namespace に作成されると、以下のようになります。

- namespace の **open-cluster-management** は、ポート 6443 で Kube API にアクセスする必要があります。

マネージドクラスターでは、**klusterlet-addon-appmgr** アプリケーションライフサイクル Pod のみが **open-cluster-management-agent-addon** namespace にインストールされます。

- namespace **open-cluster-management-agent-addon** は、ポート 6443 で Kube API にアクセスする必要があります。

- ガバナンスおよびリスク:
  ハブクラスターでは、以下のアクセスが必要です。

  - namespace の **open-cluster-management** は、ポート 6443 で Kube API にアクセスする必要があります。
  - namespace **open-cluster-management** は、ポート 5353 で OpenShift DNS にアクセスする必要があります。

  マネージドクラスターでは、以下のアクセスが必要です。

  - namespace **open-cluster-management-addon** は、ポート 6443 の Kube API にアクセスする必要があります。

1.4. SUBMARINER マルチクラスターネットワーキングおよびサービスディスカバリー

Submariner は、Red Hat Advanced Cluster Management for Kubernetes で使用できるオープンソースツールであり、オンプレミスまたはクラウドのいずれかの環境で、2 つ以上のマネージドクラスター間で直接ネットワークおよびサービスディスカバリーを提供します。Submariner は Multi-Cluster Services API (Kubernetes Enhancements Proposal #1645) と互換性があります。Submariner の詳細は、Submariner のサイト を参照してください。

どのプロバイダーが 自動コンソールデプロイメント をサポートするか、手動デプロイメント を必要とするかなど、インフラストラクチャープロバイダーのサポートレベルの詳細は、Red Hat Advanced Cluster Management サポートマトリックス を参照してください。

Submariner の使用方法の詳細は、次のトピックを参照してください。

- 非接続クラスターへの Submariner のデプロイ
- Submariner の設定
- subctl コマンドユーティリティーのインストール
- コンソールを使用した Submariner のデプロイ
- サブマリーナを手動でデプロイ
- Submariner デプロイメントのカスタマイズ
- Submariner のサービス検出の管理
- Submariner のアンインストール

1.4.1. 非接続クラスターへの Submariner のデプロイ

非接続クラスターに Submariner をデプロイすると、クラスターに対する外部からの攻撃のリスクが軽減されるため、セキュリティー上の問題を解決できます。Red Hat Advanced Cluster Management for Kubernetes を使用して Submariner を非接続クラスターにデプロイするには、非接続ネットワーク環境へのインストール で説明されている手順を最初に完了する必要があります。

1.4.1.1. 非接続クラスターで Submariner を設定する

非接続ネットワーク環境へのインストール で説明されている手順に従った後、非接続クラスターでのデプロイメントをサポートするために、インストール中に Submariner を設定する必要があります。以下のトピックを参照してください。

1.4.1.1.1. ローカルレジストリーでのイメージのミラーリング

非接続クラスターに Submariner をデプロイする前に、**Submariner Operator bundle** イメージをローカルレジストリーにミラーリングしてください。

1.4.1.1.2. catalogSource 名のカスタマイズ

デフォルトでは、**submariner-addon** は **redhat-operators** という名前の **catalogSource** を検索します。別の名前の **catalogSource** を使用する場合は、マネージドクラスターに関連付けられた **SubmarinerConfig** の **SubmarinerConfig.Spec.subscriptionConfig.Source** パラメーターの値を、**catalogSource** のカスタム名で更新する必要があります。

1.4.1.1.3. SubmarinerConfig で airGappedDeployment を有効にする

Red Hat Advanced Cluster Management for Kubernetes コンソールからマネージドクラスターに **submariner-addon** をインストールする場合、**Disconnected cluster** オプションを選択して、Submariner が外部サーバーに対して API クエリーを作成しないようにすることができます。

API を使用して Submariner をインストールする場合は、マネージドクラスターに関連付けられた **SubmarinerConfig** で **airGappedDeployment** パラメーターを **true** に設定する必要があります。

1.4.2. Submariner の設定

Red Hat Advanced Cluster Management for Kubernetes は、Submariner をハブクラスターのアドオンとして提供します。Submariner の設定方法は、次のトピックを参照してください。

- 前提条件
- Submariner ポートテーブル
- Globalnet

1.4.2.1. 前提条件

Submariner を使用する前に、以下の前提条件があることを確認します。

- **cluster-admin** のパーミッションを使用してハブクラスターにアクセスするための認証情報。
- ゲートウェイノード間で IP 接続を設定している。2 つのクラスターを接続する場合に、最低でも 1 つのクラスターには、ゲートウェイノード専用のパブリックまたはプライベート IP アドレスを使用してゲートウェイノードにアクセスできる必要があります。詳細は、**Submariner NAT Traversal** を参照してください。

- OVN Kubernetes を使用している場合には、クラスターは Red Hat OpenShift Container Platform バージョン 4.13 以降を使用する必要があります。
- Red Hat OpenShift Container Platform クラスターが OpenShift SDN CNI を使用する場合、各マネージドクラスター内のすべてのノードにわたるファイアウォール設定は、双方向で 4800/UDP を許可する必要があります。
- マネージドクラスター間のトンネルを確立するために、ファイアウォール設定では、ゲートウェイノードで 4500/UDP および 4490/UDP を許可する必要があります。
- ゲートウェイノードが間に NAT を介さずにプライベート IP 経由で直接到達できる場合は、ファイアウォール設定でゲートウェイノード上で ESP プロトコルが許可されていることを確認してください。
  注記: これは、クラスターが Amazon Web Services、Google Cloud Platform、Microsoft Azure、または Red Hat OpenStack 環境にデプロイされている場合は自動的に設定されますが、他の環境のクラスターおよびプライベートクラウドを保護するファイアウォールについては手動で設定する必要があります。
- **managedcluster** 名は、RFC 1123 で定義されている DNS ラベル標準に従い、次の要件を満たす必要があります。
  - 63 文字以内
  - 小文字の英数字またはハイフン (-) のみが含まれる。
  - 英数字で始まる。
  - 英数字で終わる。

1.4.2.2. Submariner ポートテーブル

次の表を参照して、有効にする必要のある Submariner ポートを確認してください。

名前	デフォルト値	カスタマイズ可能	任意または必須
IPsec NATT	4500/UDP	はい	必須
VXLAN	4800/UDP	いいえ	必須
NAT 検出ポート	4490/UDP	いいえ	必須

1.4.2.3. Globalnet

Globalnet は、既存のクラスターの CIDR を変更せずに、重複する Classless Inter-Domain Routings (CIDR) を使用してクラスターを接続できるようにする Submariner アドオン機能です。Globalnet は、最初のマネージドクラスターをクラスターセットに追加するときに選択できるクラスターセット全体の設定です。

Globalnet を有効にすると、すべてのマネージドクラスターは仮想グローバルプライベートネットワークからグローバル CIDR を受け取り、クラスター間の通信を容易にするために使用されます。

重要: クラスターセット内のクラスターに重複する CIDR がある可能性がある場合は、Globalnet を有効にする必要があります。

ClusterAdmin は、クラスターセット内のクラスターの Submariner アドオンを有効にするときに **Enable Globalnet** オプションを選択することで、コンソールで Globalnet を有効にできます。Globalnet を有効にした後で無効にする場合は、まずクラスターセットからすべてのマネージドクラスターを削除する必要があります。

1.4.2.3.1. submariner-broker オブジェクトを作成して Globalnet を有効にする

Red Hat Advanced Cluster Management API を使用する場合、**ClusterAdmin** は、**<ManagedClusterSet>-broker** namespace に **submariner-broker** オブジェクトを作成することで Globalnet を有効にできます。

ClusterAdmin ロールには、ブローカー名前空間に **submariner-broker** オブジェクトを作成するために必要な権限があります。クラスターセットのプロキシー管理者として機能するのに作成されることがある **ManagedClusterSetAdmin** ロールには、必要な権限がありません。

必要な権限を提供する場合は、**ClusterAdmin** が **access-to-brokers-submariner-crd** のロール権限を **ManagedClusterSetAdmin** ユーザーに関連付ける必要があります。

submariner-broker オブジェクトを作成して Globalnet を有効にするには、以下の手順を実行します。

1. 次のコマンドを実行して **<broker-namespace>** を取得します。

```
oc get ManagedClusterSet <cluster-set-name> -o jsonpath="
{.metadata.annotations['cluster\.open-cluster-management\.io/submariner-broker-ns']}"
```

2. **submariner-broker** という名前の YAML ファイルを作成して、Globalnet 設定を指定する **submariner-broker** オブジェクトを作成します。次の行のようなコンテンツを YAML ファイルに追加します。

```
apiVersion: submariner.io/v1alpha1
kind: Broker
metadata:
  name: submariner-broker 1
  namespace: broker-namespace 2
spec:
  globalnetEnabled: true-or-false 3
```

1. 名前は **submariner-broker** である必要があります。
2. **broker-namespace** を、ブローカーの namespace に置き換えます。
3. Globalnet を有効にするには、**true-or-false** を **true** に置き換えます。

3. 以下のコマンドを実行してこのファイルを適用します。

```
oc apply -f submariner-broker.yaml
```

1.4.2.3.2. グローバル IP 数の設定

ClusterGlobalEgressIP リソースの **numberOfIPs** フィールドの値を変更して、設定可能なグローバル IP のグローバル IP を割り当てることができます。デフォルト値は 8 です。以下の例を参照してください。

```
apiVersion: submariner.io/v1
kind: ClusterGlobalEgressIP
metadata:
  name: cluster-egress.submariner.io
spec:
  numberOfIPs: 8
```

1.4.2.3.3. 関連情報

- Submariner の詳細は Submariner のドキュメント を参照してください。
- ゲートウェイノード間の IP 接続に関する詳細は、Submariner NAT Traversal を参照してください。
- 前提条件の詳細は、Submariner の前提条件のドキュメント を参照してください。
- その他の使用可能なフラグの詳細は、Submariner ドキュメントの unexport を参照してください。

1.4.3. subctl コマンドユーティリティーのインストール

subctl ユーティリティーは、コンテナーイメージで提供されています。**subctl** ユーティリティーをローカルにインストールするには、次の手順を実行します。

1. 次のコマンドを実行し、プロンプトが表示されたら認証情報を入力して、レジストリーにログインします。

   ```
   oc registry login --registry registry.redhat.io
   ```

2. 次のコマンドを入力して、**subctl** コンテナー をダウンロードし、**subctl** バイナリーの圧縮バージョンを **/tmp** に展開します。

   ```
   oc image extract registry.redhat.io/rhacm2/subctl-rhel8:v0.16 --path="/dist/subctl-*-linux-amd64.tar.xz":/tmp/ --confirm
   ```

3. 次のコマンドを入力して、**subctl** ユーティリティーを展開します。

   ```
   tar -C /tmp/ -xf /tmp/subctl-v0.16*-linux-amd64.tar.xz
   ```

4. 次のコマンドを入力して、**subctl** ユーティリティーをインストールします。

   ```
   install -m744 /tmp/subctl-v0.16*/subctl-v0.16*-linux-amd64 /$HOME/.local/bin/subctl
   ```

注記:

- **subctl** と Submariner のバージョンと一致していることを確認してください。
- 非接続環境のみの場合は、**submariner-nettest** イメージをミラーリングしてください。

1.4.3.1. subctl コマンドの使用

パスにユーティリティーを追加した後に使用可能なコマンドの簡単な説明は、次の表を参照してください。

export service	指定されたサービスの **ServiceExport** リソースを作成します。これにより、Submariner デプロイメント内の他のクラスターが対応するサービスを検出できるようになります。
unexport service	指定されたサービスの **ServiceExport** リソースを削除します。これにより、Submariner デプロイメント内の他のクラスターが対応するサービスを検出できなくなります。
show	Submariner リソースに関する情報を提供します。
verify	Submariner がクラスターのペア全体で設定されている場合は、接続性、サービスディスカバリー、およびその他のサブマリーナー機能を検証します。
benchmark	Submariner で、または単一のクラスター内で有効になっているクラスターのペア全体のスループットおよびレイテンシーをベンチマークします。
diagnose	チェックを実行して、Submariner デプロイメントが正しく機能しない原因となる問題を特定します。
gather	クラスターから情報を収集して、Submariner デプロイメントのトラブルシューティングに役立てます。
version	**subctl** バイナリーツールのバージョンの詳細を表示します。

注記: **subctl** の Red Hat ビルドには、Red Hat Advanced Cluster Management for Kubernetes に関連するコマンドのみが含まれています。**subctl** ユーティリティーとそのコマンドの詳細は、Submariner ドキュメントの**subctl** を参照してください。

1.4.4. コンソールを使用した Submariner のデプロイ

Red Hat Advanced Cluster Management for Kubernetes に Submariner をデプロイする前に、ホスト環境でクラスターを準備する必要があります。**SubmarinerConfig** API または Red Hat Advanced Cluster Management for Kubernetes コンソールを使用して、以下のプロバイダーで Red Hat OpenShift Container Platform クラスターを自動的に準備できます。

- Amazon Web Services
- Google Cloud Platform
- IBM Power Systems Virtual Server
- Red Hat OpenShift on IBM Cloud (テクノロジープレビュー)
- Red Hat OpenStack Platform
- Microsoft Azure

- VMware vSphere

注記:

- VMware vSphere では、NSX 以外のデプロイメントのみがサポートされています。
- Red Hat OpenShift on IBM Cloud を使用している場合は、クラスターに Calico API サーバー をインストールする必要があります。あるいは、Submariner アップストリームドキュメントの CALICO CNI トピックに従って、クラスター間通信に必要な IP プールを手動で作成することもできます。

他のプロバイダーに Submariner をデプロイするには、Submariner の手動デプロイ を参照してください。

Red Hat Advanced Cluster Management for Kubernetes コンソールで Submariner をデプロイするには、以下の手順を実行します。

必要なアクセス権限: クラスターの管理者

1. コンソールで、**Infrastructure** > **Clusters** を選択します。
2. **Clusters** ページで、**Cluster sets** タブを選択します。Submariner で有効にするクラスターは、同じクラスターセットにある必要があります。
3. Submariner をデプロイするクラスターがすでに同じクラスターセットにある場合は、手順 5 に進みます。
4. Submariner をデプロイするクラスターが同じクラスターセットにない場合は、以下の手順に従ってクラスターセットを作成します。
   a. **Create cluster set**を選択します。
   b. クラスターセットに名前を付け、**Create** を選択します。
   c. **Manage resource assignments** を選択して、クラスターセットに割り当てます。
   d. Submariner で接続するマネージドクラスターを選択して、クラスターセットに追加します。
   e. **Review** を選択して、選択したクラスターを表示し、確認します。
   f. **Save** を選択してクラスターセットを保存し、作成されるクラスターセットページを表示します。
5. クラスターセットページで、**Submariner add-on** タブを選択します。
6. **Install Submariner add-ons** を選択します。
7. Submariner をデプロイするクラスターを選択します。
8. 次の表のフィールドを参照し、**Install Submariner アドオン** エディターに必要な情報を入力します。

フィールド	注記
AWS Access Key ID	AWS クラスターをインポートする場合にのみ表示されます。

フィールド	注記
AWS Secret Access Key	AWS クラスターをインポートする場合にのみ表示されます。
Base domain resource group name	Azure クラスターをインポートする場合にのみ表示されます。
Client ID	Azure クラスターをインポートする場合にのみ表示されます。
クライアントシークレット	Azure クラスターをインポートする場合にのみ表示されます。
サブスクリプション ID	Azure クラスターをインポートする場合にのみ表示されます。
テナント ID	Azure クラスターをインポートする場合にのみ表示されます。
Google Cloud Platform service account JSON key	Google Cloud Platform クラスターをインポートする場合にのみ表示されます。
インスタンスタイプ	マネージドクラスターで作成されるゲートウェイノードのインスタンスタイプ。
IPsec NAT-T port	IPsec NAT トラバーサルポートのデフォルト値はポート **4500** です。マネージドクラスター環境が VMware vSphere の場合は、ファイアウォールでこのポートが開いていることを確認してください。
ゲートウェイ数	マネージドクラスターにデプロイされるゲートウェイノードの数。AWS、GCP、Azure、および OpenStack クラスターの場合、専用のゲートウェイノードがデプロイされます。VWware クラスターの場合、既存のワーカーノードはゲートウェイノードとしてタグ付けされます。デフォルト値は **1** です。値が 1 を超える場合、Submariner ゲートウェイの High Availability (HA) は自動的に有効になります。
ケーブルドライバー	クラスター間トンネルを維持する Submariner ゲートウェイケーブルエンジンのコンポーネントです。デフォルト値は **Libreswan IPsec** です。
Disconnected cluster	有効にすると、パブリック IP 解決のために外部サーバーにアクセスしないように Submariner に指示します。

フィールド	注記
Globalnet CIDR	クラスターセットで Globalnet 設定が選択されている場合にのみ表示されます。マネージドクラスターに使用される Globalnet CIDR。空白のままにすると、クラスターセットプールから CIDR が割り当てられます。

9. エディターの末尾で **Next** を選択して、次のクラスターのエディターに移動し、選択した残りのクラスターごとに、エディターを完了します。

10. 各マネージドクラスターの設定を確認します。

11. **Install** をクリックして、選択したマネージドクラスターに Submariner をデプロイします。インストールと設定が完了するまで数分かかる場合があります。**Submariner add-on** タブのリストで Submariner ステータスを確認できます。

    - **Connection status** は、マネージドクラスターで確立される Submariner 接続の数を示します。
    - **Agent status** は、Submariner がマネージドクラスターに正常にデプロイされるかどうかを示します。コンソールでは、インストールと設定が完了するまで **Degraded** のステータスをレポートする場合があります。
    - **Gateway nodes labeled** はマネージドクラスター上のゲートウェイノードの数を示します。

Submariner が選択したクラスターにデプロイされました。

1.4.5. サブマリーナを手動でデプロイ

Red Hat Advanced Cluster Management for Kubernetes に Submariner をデプロイする前に、接続用にホスト環境でクラスターを準備する必要があります。コンソールを使用して Submariner をサポートされているクラスターに自動的にデプロイする方法は、コンソールを使用して Submariner をデプロイする を参照してください。

Submariner の自動デプロイメントをサポートしていないプロバイダーでクラスターがホスティングされている場合は、次のセクションを参照してインフラストラクチャーを手動で準備してください。プロバイダーごとに固有の準備手順があるため、正しいプロバイダーを選択してください。

1.4.5.1. Submariner 向けのベアメタルの準備

Submariner をデプロイするためのベアメタルクラスターを準備するには、次の手順を実行します。

1. ファイアウォールが、ゲートウェイノードの 4500/UDP ポートおよび 4490/UDP ポートで外部クライアントの受信/送信トラフィックを許可していることを確認します。また、クラスターが OpenShiftSDN CNI を使用してデプロイされている場合は、ローカルクラスターノード内のインバウンド/アウトバウンド UDP/4800 トラフィックを許可します。

2. 次の例のような YAML コンテンツをカスタマイズして適用します。

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
```

```
    name: submariner
    namespace: <managed-cluster-namespace>
spec:
    gatewayConfig:
      gateways: 1
```

managed-cluster-namespace をマネージドクラスターの名前に置き換えます。以下の例のように、**SubmarinerConfig** の名前は **submariner** である必要があります。

この設定では、ワーカーノードの 1 つをベアメタルクラスターの Submariner ゲートウェイとしてラベル付けします。

デフォルトでは、Submariner は IP セキュリティー (IPsec) を使用して、ゲートウェイノード上のクラスター間でセキュアなトンネルを確立します。デフォルトの IPsec NATT ポートを使用するか、設定した別のポートを指定できます。IPsec NATT ポートを指定せずに、この手順を実行すると、接続に 4500/UDP が使用されます。

3. Submariner によって設定されたゲートウェイノードを特定し、ファイアウォール設定を有効にして、外部トラフィック用の IPsec NATT (UDP/4500) および NatDiscovery (UDP/4490) ポートを許可します。

カスタマイズオプションは、Submariner デプロイメントのカスタマイズ を参照してください。

1.4.5.2. コマンドラインインターフェイスを使用した Microsoft Azure Red Hat OpenShift for Submariner の準備

Microsoft Azure Red Hat OpenShift サービスは、コンテナーベースのアプリケーションの構築プロセスを簡素化するために使用できるさまざまなツールとリソースを組み合わせています。コマンドラインインターフェイスを使用して Submariner をデプロイするために Azure Red Hat OpenShift クラスターを準備するには、次の手順を実行します。

1. Azure CLI をインストールします。
2. Azure CLI から、次のコマンドを実行して拡張機能をインストールします。

```
az extension add --upgrade -s <path-to-extension>
```

.whl 拡張ファイルをダウンロードした場所へのパスに **path-to-extension** を置き換えます。

3. 次のコマンドを実行して、CLI 拡張機能が使用されていることを確認します。

```
az extension list
```

拡張機能が使用されている場合、出力は次の例のようになります。

```
"experimental": false,
"extensionType": "whl",
"name": "aro",
"path": "<path-to-extension>",
"preview": true,
"version": "1.0.x"
```

4. Azure CLI から、次のコマンドを実行してプレビュー機能を登録します。

```
az feature registration create --namespace Microsoft.RedHatOpenShift --name
AdminKubeconfig
```

5. 次のコマンドを実行して、管理者 **kubeconfig** を取得します。

```
az aro get-admin-kubeconfig -g <resource group> -n <cluster resource name>
```

注記: az aro コマンドは、**kubeconfig** をローカルディレクトリーに保存し、**kubeconfig** という名前を使用します。これを使用するには、環境変数 **KUBECONFIG** をファイルのパスと一致するように設定します。以下の例を参照してください。

```
export KUBECONFIG=<path-to-kubeconfig>
oc get nodes
```

6. Azure Red Hat OpenShift クラスターをインポートします。クラスターをインポートする方法の詳細は、クラスターのインポートの概要 を参照してください。

1.4.5.2.1. API を使用した Microsoft Azure Red Hat OpenShift for Submariner の準備

API を使用して Submariner をデプロイするために Azure Red Hat OpenShift クラスターを準備するには、次の例のような YAML コンテンツをカスタマイズして適用します。

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: submariner
  namespace: <managed-cluster-namespace>
spec:
  loadBalancerEnable: true
```

managed-cluster-namespace をマネージドクラスターの名前に置き換えます。

以下の例のように、**SubmarinerConfig** の名前は **submariner** である必要があります。

この設定では、ワーカーノードの 1 つを Azure Red Hat OpenShift クラスターの Submariner ゲートウェイとしてラベル付けします。

デフォルトでは、Submariner は IP セキュリティー (IPsec) を使用して、ゲートウェイノード上のクラスター間でセキュアなトンネルを確立します。デフォルトの IPsec NATT ポートを使用するか、設定した別のポートを指定できます。IPsec NATT ポートを指定せずに、この手順を実行すると、接続にポート 4500/UDP が使用されます。

カスタマイズオプションは、Submariner デプロイメントのカスタマイズ を参照してください。

1.4.5.3. コマンドラインインターフェイスを使用した Submariner 用の Red Hat OpenShift Service on AWS の準備

Red Hat OpenShift Service on AWS は、アプリケーションの開発と最新化のための安定した柔軟なプラットフォームを提供します。Submariner をデプロイするために OpenShift Service on AWS クラスターを準備するには、次の手順を実行します。

1. 次のコマンドを実行して、OpenShift Service on AWS にログインします。

```
rosa login
oc login <rosa-cluster-url>:6443 --username cluster-admin --password <password>
```

2. 次のコマンドを実行して、OpenShift Service on AWS クラスターの **kubeconfig** を作成します。

```
oc config view --flatten=true > rosa_kube/kubeconfig
```

3. OpenShift Service on AWS クラスターをインポートします。クラスターをインポートする方法の詳細は、クラスターのインポートの概要 を参照してください。

1.4.5.3.1. API を使用した Submariner 用の Red Hat OpenShift Service on AWS の準備

API を使用して Submariner をデプロイするために OpenShift Service on AWS クラスターを準備するには、次の例のような YAML コンテンツをカスタマイズして適用します。

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: submariner
  namespace: <managed-cluster-namespace>
spec:
  loadBalancerEnable: true
```

managed-cluster-namespace をマネージドクラスターの名前に置き換えます。

以下の例のように、**SubmarinerConfig** の名前は **submariner** である必要があります。

デフォルトでは、Submariner は IP セキュリティー (IPsec) を使用して、ゲートウェイノード上のクラスター間でセキュアなトンネルを確立します。デフォルトの IPsec NATT ポートを使用するか、設定した別のポートを指定できます。IPsec NATT ポートを指定せずに、この手順を実行すると、接続にポート 4500/UDP が使用されます。

カスタマイズオプションは、Submariner デプロイメントのカスタマイズ を参照してください。

1.4.5.4. ManagedClusterAddOn API を使用した Submariner のデプロイ

選択したホスティング環境を手動で準備した後、次の手順を完了することで、**ManagedClusterAddOn** API を使用して Submariner をデプロイできます。

1. ManagedClusterSet の作成 ドキュメントに記載されている手順を使用して、ハブクラスターに **ManagedClusterSet** リソースを作成します。**ManagedClusterSet** のエントリーが次の内容のようになっていることを確認してください。

```
apiVersion: cluster.open-cluster-management.io/v1beta2
kind: ManagedClusterSet
metadata:
  name: <managed-cluster-set-name>
```

managed-cluster-set-name は、作成する **ManagedClusterSet** の名前に置き換えます。

重要: Kubernetes namespace の最大文字数は 63 文字です。 **<managed-cluster-set-name>** に使用できる最大文字数は 56 文字です。**<managed-cluster-set-name>** の文字数が 56 文字を超える場合、**<managed-cluster-set-name>** は先頭から切り捨てられます。

ManagedClusterSet が作成されたら、**submariner-addon** は **<managed-cluster-set-name>-broker** と呼ばれる namespace を作成し、その namespace に Submariner ブローカーをデプロイします。

2. 次の例のような YAML コンテンツをカスタマイズして適用することにより、**<managed-cluster-set-name>-broker** namespace のハブクラスターに **Broker** 設定を作成します。

```
apiVersion: submariner.io/v1alpha1
kind: Broker
metadata:
    name: submariner-broker
    namespace: <managed-cluster-set-name>-broker
    labels:
       cluster.open-cluster-management.io/backup: submariner
spec:
    globalnetEnabled: <true-or-false>
```

managed-cluster-set-name は、マネージドクラスターの名前に置き換えます。

ManagedClusterSet で Submariner Globalnet を有効にする場合は、 **globalnetEnabled** の値を **true** に設定します。

3. 次のコマンドを実行して、1 つのマネージドクラスターを **ManagedClusterSet** に追加します。

```
oc label managedclusters <managed-cluster-name> "cluster.open-cluster-
management.io/clusterset=<managed-cluster-set-name>" --overwrite
```

managedcluster-name は、**ManagedClusterSet** に追加するマネージドクラスターの名前に置き換えます。

ManagedClusterSet-name は、マネージドクラスターを追加する **ManagedClusterSet** の名前に置き換えます。

4. 次の例のような YAML コンテンツをカスタマイズして適用します。

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
   name: submariner
   namespace: <managed-cluster-namespace>
spec:{}
```

managed-cluster-namespace は、マネージドクラスターの namespace に置き換えます。

注記: 以下の例のように、**SubmarinerConfig** の名前は **submariner** である必要があります。

5. 次の例のような YAML コンテンツをカスタマイズして適用することにより、マネージドクラスターに Submariner をデプロイします。

```
apiVersion: addon.open-cluster-management.io/v1alpha1
kind: ManagedClusterAddOn
metadata:
    name: submariner
    namespace: <managed-cluster-name>
spec:
    installNamespace: submariner-operator
```

managedcluster-name は、Submariner で使用するマネージドクラスターの名前に置き換えます。

ManagedClusterAddOn の仕様の **installNamespace** フィールドは、Submariner をインストールするマネージドクラスター上の namespace に置き換えます。現在、**Submariner-operator** namespace に Submariner をインストールする必要があります。

ManagedClusterAddOn の作成後に、**submariner-addon** は Submariner をマネージドクラスターの **submariner-operator** namespace にデプロイします。この **ManagedClusterAddOn** のステータスから Submariner のデプロイメントステータスを表示できます。

注記: **ManagedClusterAddOn** の名前は **submariner** である必要があります。

6. Submariner を有効にするすべてのマネージドクラスターに対して、手順 3、4、および 5 を繰り返します。

7. マネージドクラスターに Submariner をデプロイしたら、次のコマンドを入力して、Submariner **ManagedClusterAddOn** のステータスを確認して、Submariner のデプロイメントステータスを確認できます。

```
oc -n <managed-cluster-name> get managedclusteraddons submariner -oyaml
```

cluster-name は、マネージドクラスターの名前に置き換えます。

Submariner **ManagedClusterAddOn** のステータスの 3 つの条件により、Submariner のデプロイメントステータスが分かります。

- **SubmarinerGatewayNodesLabeled** の条件は、マネージドクラスターに Submariner ゲートウェイノードにラベル付けされているかどうかを示します。
- **SubmarinerAgentDegraded** の条件は、Submariner がマネージドクラスターに正常にデプロイされるかどうかを示します。
- **SubmarinerConnectionDegraded** の条件は、Submariner でマネージドクラスターで確立される接続の数を示します。

1.4.6. Submariner デプロイメントのカスタマイズ

NATT (Network Address Translation-Traversal) ポート、ゲートウェイノードの数、ゲートウェイノードのインスタンスタイプなど、Submariner デプロイメントの設定の一部をカスタマイズできます。これらのカスタマイズは、すべてのプロバイダーで一貫しています。

1.4.6.1. NATT ポート

NATT ポートをカスタマイズする場合は、プロバイダー環境に合わせて次の YAML コンテンツをカスタマイズして適用します。

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
   name: submariner
   namespace: <managed-cluster-namespace>
spec:
```

```
  credentialsSecret:
    name: <managed-cluster-name>-<provider>-creds
  IPSecNATTPort: <NATTPort>
```

- **managed-cluster-namespace** は、マネージドクラスターの namespace に置き換えます。
- **managed-cluster-name** は、マネージドクラスターの名前に置き換えます。
  - AWS: **provider** を **aws** に置き換えます。**<managed-cluster-name>-aws-creds** の値は、AWS の認証情報シークレット名で、この情報はハブクラスターのクラスター namespace にあります。
  - GCP: **provider** を **gcp** に置き換えます。**<managed-cluster-name>-gcp-creds** の値は、Google Cloud Platform 認証情報シークレット名を指し、ハブクラスターのクラスター namespace で見つけることができます。
  - OpenStack: **provider** を **osp** に置き換えます。**<managed-cluster-name>-osp-creds** の値は、ハブクラスターのクラスター namespace にある Red Hat OpenStack Platform 認証情報シークレット名です。
  - Azure: **provider** を **azure** に置き換えます。**<managed-cluster-name>-azure-creds** の値は、ハブクラスターのクラスター namespace で見つけることができる Microsoft Azure 認証情報シークレット名です。
- **managed-cluster-namespace** は、マネージドクラスターの namespace に置き換えます。
- **managed-cluster-name** は、マネージドクラスターの名前に置き換えます。**managed-cluster-name-gcp-creds** の値は、Google Cloud Platform 認証情報シークレット名を指し、ハブクラスターのクラスター namespace で見つけることができます。
- **NATTPort** は、使用する NATT ポートに置き換えます。

注記: 以下の例のように、**SubmarinerConfig** の名前は **submariner** である必要があります。

1.4.6.2. ゲートウェイノードの数

ゲートウェイノードの数をカスタマイズする場合は、次の例のような YAML コンテンツをカスタマイズして適用します。

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: submariner
  namespace: <managed-cluster-namespace>
spec:
  credentialsSecret:
    name: <managed-cluster-name>-<provider>-creds
  gatewayConfig:
    gateways: <gateways>
```

- **managed-cluster-namespace** は、マネージドクラスターの namespace に置き換えます。
- **managed-cluster-name** は、マネージドクラスターの名前に置き換えます。

- AWS: **provider** を **aws** に置き換えます。**<managed-cluster-name>-aws-creds** の値は、AWS の認証情報シークレット名で、この情報はハブクラスターのクラスター namespace にあります。
- GCP: **provider** を **gcp** に置き換えます。**<managed-cluster-name>-gcp-creds** の値は、Google Cloud Platform 認証情報シークレット名を指し、ハブクラスターのクラスター namespace で見つけることができます。
- OpenStack: **provider** を **osp** に置き換えます。**<managed-cluster-name>-osp-creds** の値は、ハブクラスターのクラスター namespace にある Red Hat OpenStack Platform 認証情報シークレット名です。
- Azure: **provider** を **azure** に置き換えます。**<managed-cluster-name>-azure-creds** の値は、ハブクラスターのクラスター namespace で見つけることができる Microsoft Azure 認証情報シークレット名です。

- **gateways** は、使用するゲートウェイ数に置き換えます。値が 1 より大きい場合には、Submariner ゲートウェイは高可用性を自動的に有効にします。

注記: 以下の例のように、**SubmarinerConfig** の名前は **submariner** である必要があります。

1.4.6.3. ゲートウェイノードのインスタンスタイプ

ゲートウェイノードのインスタンスタイプをカスタマイズする場合は、次の例のような YAML コンテンツをカスタマイズして適用します。

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: submariner
  namespace: <managed-cluster-namespace>
spec:
  credentialsSecret:
    name: <managed-cluster-name>-<provider>-creds
 gatewayConfig:
    instanceType: <instance-type>
```

- **managed-cluster-namespace** は、マネージドクラスターの namespace に置き換えます。
- **managed-cluster-name** は、マネージドクラスターの名前に置き換えます。
  - AWS: **provider** を **aws** に置き換えます。**<managed-cluster-name>-aws-creds** の値は、AWS の認証情報シークレット名で、この情報はハブクラスターのクラスター namespace にあります。
  - GCP: **provider** を **gcp** に置き換えます。**<managed-cluster-name>-gcp-creds** の値は、Google Cloud Platform 認証情報シークレット名を指し、ハブクラスターのクラスター namespace で見つけることができます。
  - OpenStack: **provider** を **osp** に置き換えます。**<managed-cluster-name>-osp-creds** の値は、ハブクラスターのクラスター namespace にある Red Hat OpenStack Platform 認証情報シークレット名です。
  - Azure: **provider** を **azure** に置き換えます。**<managed-cluster-name>-azure-creds** の値は、ハブクラスターのクラスター namespace で見つけることができる Microsoft Azure 認証情報シークレット名です。

- **instance-type** は、使用する AWS インスタンスタイプに置き換えます。

注記: 以下の例のように、**SubmarinerConfig** の名前は **submariner** である必要があります。

1.4.6.4. ケーブルドライバー

Submariner Gateway Engine コンポーネントは、他のクラスターへの安全なトンネルを作成します。ケーブルドライバーコンポーネントは、ゲートウェイエンジンコンポーネントのプラグ可能なアーキテクチャーを使用してトンネルを維持します。ケーブルエンジンコンポーネントの **cableDriver** 設定には、Libreswan または VXLAN 実装を使用できます。以下の例を参照してください。

```
apiVersion: submarineraddon.open-cluster-management.io/v1alpha1
kind: SubmarinerConfig
metadata:
  name: submariner
  namespace: <managed-cluster-namespace>
spec:
  cableDriver: vxlan
  credentialsSecret:
    name: <managed-cluster-name>-<provider>-creds
```

ベストプラクティス: パブリックネットワークでは VXLAN ケーブルドライバーを使用しないでください。VXLAN ケーブルドライバーは暗号化されていません。プライベートネットワークでの不要な二重暗号化を避けるために、VXLAN のみを使用してください。たとえば、一部のオンプレミス環境では、専用の回線レベルのハードウェアデバイスを使用してトンネルの暗号化を処理する場合があります。

1.4.6.5. カスタマイズされた Submariner サブスクリプションの使用

Submariner アドオンは、Submariner のサブスクリプションを自動的に設定し、インストールされている Red Hat Advanced Cluster Management のバージョンに適切な Submariner のバージョンがインストールされ、最新の状態に保たれます。この動作を変更する場合、または Submariner のアップグレードを手動で制御する場合は、Submariner サブスクリプションをカスタマイズできます。

カスタマイズされた Submariner サブスクリプションを使用する場合は、次のフィールドに入力する必要があります。

- **Source:** Submariner サブスクリプションに使用するカタログソース。たとえば、**redhat-operators** です。
- **Source Namespace:** カタログソースの namespace。たとえば、**openshift-marketplace** などです。
- **Channel:** サブスクリプション用にフォローするチャンネル。たとえば、Red Hat Advanced Cluster Management 2.9 **stable-0.16** の場合は、以下のようになります。
- **Starting CSV (オプション):** 初期の **ClusterServiceVersion**。
- **Install Plan Approval:** インストール計画を手動で承認するか、または自動的に承認するか。

注: インストールプランを手動で承認する場合は、カスタマイズされた Submariner サブスクリプションを使用する必要があります。

1.4.7. Submariner のサービス検出の管理

Submariner がマネージドクラスターと同じ環境にデプロイされた後、マネージドクラスターセット内のクラスター全体で Pod とサービス間の安全な IP ルーティングのためにルートが設定されます。

1.4.7.1. Submariner のサービス検出の有効化

クラスターからのサービスをマネージドクラスターセット内の他のクラスターに表示および検出可能にするには、**ServiceExport** オブジェクトを作成する必要があります。**ServiceExport** オブジェクトでサービスをエクスポートすると、**<service>.<namespace>.svc.clusterset.local** 形式でサービスにアクセスできます。複数のクラスターが同じ名前で、同じ namespace からサービスをエクスポートすると、他のクラスターは、その複数のクラスターを 1 つの論理サービスとして認識します。

この例では、**default** の namespace で **nginx** サービスを使用しますが、Kubernetes の **ClusterIP** サービスまたはヘッドレスサービスを検出できます。

1. 以下のコマンドを入力して、**ManagedClusterSet** のマネージドクラスターに **nginx** サービスのインスタンスを適用します。

```
oc -n default create deployment nginx --image=nginxinc/nginx-unprivileged:stable-alpine
oc -n default expose deployment nginx --port=8080
```

2. 次のコマンドのような **subctl** ツールを使用してコマンドを入力し、**ServiceExport** エントリーを作成して、サービスをエクスポートします。

```
subctl export service --namespace <service-namespace> <service-name>
```

   service-namespace を、サービスが置かれた namespace の名前に置き換えます。この例では、**default** になります。

   service-name を、エクスポートするサービスの名前に置き換えます。この例では、**nginx** になります。

   その他の使用可能なフラグの詳細は、Submariner ドキュメントの **export** を参照してください。

3. 別のマネージドクラスターから以下のコマンドを実行して、**nginx** サービスにアクセスできることを確認します。

```
oc -n default run --generator=run-pod/v1 tmp-shell --rm -i --tty --image
quay.io/submariner/nettest -- /bin/bash curl nginx.default.svc.clusterset.local:8080
```

これで、**nginx** サービス検出が Submariner に対して設定されました。

1.4.7.2. Submariner のサービス検出の無効化

サービスが他のクラスターにエクスポートされないようにするには、**nginx** の次の例のようなコマンドを入力します。

```
subctl unexport service --namespace <service-namespace> <service-name>
```

service-namespace を、サービスが置かれた namespace の名前に置き換えます。

service-name を、エクスポートするサービスの名前に置き換えます。

その他の使用可能なフラグの詳細は、Submariner ドキュメントの **unexport** を参照してください。

このサービスは、クラスターによる検出に使用できなくなりました。

1.4.8. Submariner のアンインストール

Red Hat Advanced Cluster Management for Kubernetes コンソールまたはコマンドラインを使用して、クラスターから Submariner コンポーネントをアンインストールできます。0.12 より前の Submariner バージョンで、すべてのデータプレーンコンポーネントを完全に削除するには、追加の手順が必要です。Submariner のアンインストールはべき等であるため、問題なく手順を繰り返すことができます。

1.4.8.1. コンソールを使用した Submariner のアンインストール

コンソールを使用してクラスターから Submariner をアンインストールするには、次の手順を実行します。

1. コンソールナビゲーションから、**Infrastructure** > **Clusters** を選択し、**Cluster sets** タブを選択します。
2. Submariner コンポーネントを削除するクラスターを含むクラスターセットを選択します。
3. **Submariner Add-ons** タブを選択して、Submariner がデプロイされているクラスターセット内のクラスターを表示します。
4. Submariner をアンインストールするクラスターの **Actions** メニューで、**Uninstall Add-on** を選択します。
5. Submariner をアンインストールするクラスターの **アクション** メニューで、**クラスターセットの削除** を選択します。
6. Submariner を削除する他のクラスターについても、これらの手順を繰り返します。
   ヒント: 複数のクラスターを選択して **Actions** をクリックすると、同じクラスターセット内の複数のクラスターから Submariner アドオンを削除できます。**Uninstall Submariner add-ons** を選択します。

削除する Submariner のバージョンがバージョン 0.12 より前の場合は、Submariner を手動でアンインストールする に進みます。Submariner のバージョンが 0.12 以降の場合、Submariner は削除されます。

重要: クラウドプロバイダーによる追加料金を回避するために、すべてのクラウドリソースがクラウドプロバイダーから削除されていることを確認してください。詳細は、Submariner リソースの削除の確認 を参照してください。

1.4.8.2. CLI を使用した Submariner のアンインストール

コマンドラインを使用して Submariner をアンインストールするには、次の手順を実行します。

1. 次のコマンドを実行して、クラスターの Submariner デプロイメントを削除します。

   ```
   oc -n <managed-cluster-namespace> delete managedclusteraddon submariner
   ```

   managed-cluster-namespace は、マネージドクラスターの namespace に置き換えます。

2. 次のコマンドを実行して、クラスターのクラウドリソースを削除します。

   ```
   oc -n <managed-cluster-namespace> delete submarinerconfig submariner
   ```

managed-cluster-namespace は、マネージドクラスターの namespace に置き換えます。

3. 次のコマンドを実行して、クラスターセットを削除し、ブローカーの詳細を削除します。

```
oc delete managedclusterset <managedclusterset>
```

managedclusterset をマネージドクラスターセットの名前に置き換えます。

削除する Submariner のバージョンがバージョン 0.12 より前の場合は、Submariner を手動でアンインストールする に進みます。Submariner のバージョンが 0.12 以降の場合、Submariner は削除されます。

重要: クラウドプロバイダーによる追加料金を回避するために、すべてのクラウドリソースがクラウドプロバイダーから削除されていることを確認してください。詳細は、Submariner リソースの削除の確認 を参照してください。

1.4.8.3. Submariner の手動アンインストール

バージョン 0.12 より前のバージョンの Submariner をアンインストールする場合は、Submariner ドキュメントの 手動アンインストール セクションの手順 5 ~ 8 を実行してください。

これらの手順を完了すると、Submariner コンポーネントがクラスターから削除されます。

重要: クラウドプロバイダーによる追加料金を回避するために、すべてのクラウドリソースがクラウドプロバイダーから削除されていることを確認してください。詳細は、Submariner リソースの削除の確認 を参照してください。

1.4.8.4. Submariner リソースの削除の確認

Submariner をアンインストールした後、すべての Submariner リソースがクラスターから削除されていることを確認します。それらがクラスターに残っている場合、一部のリソースはインフラストラクチャープロバイダーからの料金を引き続き発生させます。次の手順を実行して、クラスターに追加の Submariner リソースがないことを確認します。

1. 次のコマンドを実行して、クラスターに残っている Submariner リソースをリスト表示します。

```
oc get cluster <CLUSTER_NAME> grep submariner
```

CLUSTER_NAME をクラスターの名前に置き換えます。

2. 次のコマンドを入力して、リストのリソースをすべて削除します。

```
oc delete resource <RESOURCE_NAME> cluster <CLUSTER_NAME>
```

RESOURCE_NAME を、削除する Submariner リソースの名前に置き換えます。

3. 検索でリソースが特定されなくなるまで、クラスターごとに手順 1 ~ 2 を繰り返します。

Submariner リソースがクラスターから削除されます。